KEKER & VAN NEST LLP

Robert Van Nest
rvannest@kvn.com

November 4, 2011

The Honorable William Alsup
United States District Court, Northern District of California
450 Golden Gate Avenue
San Francisco, California 94102

Re: *Oracle America, Inc. v. Google Inc*., No. 3:10-CV-03561-WHA (N.D. Cal.)

Dear Judge Alsup:

In its October 31 Order, the Court denied both parties' requests to file motions, *except* that the Court granted Google leave to file "a more developed précis . . . . that more fully develops its request for an advance determination regarding the 37 API specifications and the issue of their selection, arrangement, and structure." 10/31/11 Order (Dkt. 584) at 2. That is, by its "more developed précis" Google can request an advance determination of whether any of the allegedly copied aspects of the selection, arrangement and structure are copyrightable.

I. Determining copyrightability does not require a jury.

As Google explained in its prior précis, an advance determination is possible because determining copyrightability does not require fact finding by a jury. Instead, "[u]sing analytic dissection . . . *the court* must determine whether any of the allegedly similar features are protected by copyright." *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 (9th Cir. 1994) *(Apple VII)* (emphasis added). Oracle has conceded that copyrightability is a question of law. *See, e.g.*, 10/27/11 Jacobs Ltr. (Dkt. 566) at 3. Moreover, "issues of copyrightability, *including any fact questions bearing upon them*, must be determined by the court, not the jury." *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F. Supp. 78, 96 (D. Mass. 1992) (emphasis added); *Pivot Point, Int'l, Inc. v. Charlene Prods., Inc.*, 932 F. Supp. 220, 225 (N.D. Ill. 1996) (Easterbrook, J., sitting by designation) ("[a] jury has nothing to do with" the copyrightability determination, citing *Markman v. Westview Instruments, Inc.*, 517 U.S. 370 (1996)).

Oracle's suggestion that there may be "threshold factual determinations" for the jury (*see* 10/27/11 Jacobs Ltr. at 3) is unfounded. *First*, nowhere in its proposed verdict form (Dkt. 531-1) does Oracle ask the jury to make any such threshold factual determinations. Instead, Oracle

states in its proposed jury instructions that "upon the close of evidence, the Court should instruct the jury as to which works are protected by copyright . . . ." Joint Proposed Jury Instructions (Dkt. 539) at 54. Thus Oracle proposes that the Court decide copyrightability without first having the jury make any threshold factual determinations.

Second, the authority Oracle cites in support of this suggestion is incorrect. *See* 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 12.10[B][1]. The sole authority the Nimmer treatise cites is *Montgomery v. Noga*, 168 F.3d 1282, 1291 n.14 (11th Cir. 1999). Although there was a jury trial in that case, *see id.* at 1286, the Eleventh Circuit explained that it was reviewing a factual finding by *the district court*, and that this review was for clear error, even though *the district court's* determination did not rest on credibility determinations. *Id.* at 1291 n.14 (citing *Anderson v. City of Bessemer City*, 470 U.S. 564, 574 (1985)). Far from demonstrating that threshold factual issues must be decided by the jury, *Montgomery* confirms that fact questions bearing on issues of copyrightability must be decided by the court, even in cases tried to a jury.

II. *Apple Computer v. Microsoft Corp.* provides a roadmap for a copyrightability determination.

Of the six published district court decisions in *Apple Computer v. Microsoft Corp.*, the fifth and sixth are the most instructive for present purposes. In that case, Apple alleged that Windows copied the "look and feel" of the Macintosh user interface, and thereby infringed Apple's copyrights. *See Apple VII*, 35 F.3d at 1438. In the fifth district court decision, Judge Walker held that the "look and feel" of Apple's interface as a whole was not protectable separate and apart from the individual elements of the interface. *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1022-23 (N.D. Cal. 1992) *(Apple V)*. The court reasoned that "[t]he very purpose of a computer user interface lies in its ability to help people prepare and analyze their work quickly and flexibly." *Id.* at 1023 (quotation marks and citation omitted). Under those circumstances, the court held that copyright protection could "attach only to such a

product's separate artistic features or can afford only such limited protection as appropriate when its features are the product of a compilation." *Id.* (citing *Harper House, Inc. v. Thomas Nelson, Inc.*, 889 F.2d 197, 205 (9th Cir. 1989)). The court also considered the allegedly infringed elements that Apple had identified, and concluded that most of them were not protectable. *Apple V*, 799 F. Supp. at 1026-47. In so doing, the court considered, among other doctrines, the idea/expression dichotomy, merger, functionality, and *scenes a faire*. *Id.* at 1021-22.

In the sixth decision, the court granted partial summary judgment of non-infringement, and identified several issues that remained to be tried. *Apple Computer, Inc. v. Microsoft Corp.*, 821 F. Supp. 616, 619-22, 631 (N.D. Cal. 1993) *(Apple VI)*. The defendants then moved for summary judgment on those issues, which Apple did not oppose. *Apple VII*, 35 F.3d at 1438. On appeal, the Ninth Circuit affirmed, approved the procedures followed by the district court, and set forth the "helpful" steps to be followed in cases such as this. *Id.* at 1443.

A. Oracle should identify the allegedly copied elements.

To ensure an orderly and efficient presentation of the case to the jury, it is crucial to know what it is that Oracle is claiming was copied. The Court should direct Oracle to explain, precisely, what Google allegedly copied from the selection, arrangement and structure of Oracle's 37 API specifications. *See id*. In the *Apple Computer* district court proceedings, "[p]ursuant to Judge Schwarzer's direction, Apple identified 189 Macintosh visual displays which it claimed appear in Windows Version 2.03 and NewWave." *Apple Computer, Inc. v. Microsoft Corp.*, 759 F. Supp. 1444, 1449 (N.D. Cal. 1991) *(Apple III)*. Here, Oracle should be required to identify the aspects of the selection, arrangement and structure of the 37 API specifications on which it bases its claim.

For example, with respect to the accused API specifications, Oracle now appears to be relying on the selection of certain API elements ("classes, subclasses, interfaces, fields, methods, error descriptions and exceptions") that are described by the 37 API specifications, and their

"hierarchy." *See* 10/27/11 Jacobs Ltr. (Dkt. 566) at 1. It also apparently relies on certain "interrelationships," but does not explain what that means. *See id.* And it apparently further relies on the "layout" of the API specifications, but it is again unclear what this means, and how it differs, if at all, from the selection and hierarchy. *See id.* To the extent that Oracle is referring to the visual "layout" of the API specifications (for example, the fact that terms being described appear in the left column, and the text descriptions appear in the right column), Oracle has not identified the visual elements of its layout that it claims have been copied. Moreover, it is unclear how any such aspects of the "layout" are any more creatively expressive than the layout of the telephone directory that the Supreme Court concluded was not copyrightable in *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991); *see also Matthew Bender & Co., Inc. v. West Pub. Co.*, 158 F.3d 674, 682 (2d Cir. 1998) (finding "[t]he creative spark is missing where . . . the author made obvious, garden-variety, or routine selections . . . .").

The metes and bounds of Oracle's claim that the Android *source code* implementing the APIs described by the 37 API packages is an infringing derivative work are similarly vague and undefined. Oracle is not claiming that the code that *implements* the APIs infringes. *See* 9/15/11 Tr. at 51 ("We are not claiming that this code is not an—that this code is not an independent implementation."). Instead, it appears to be claiming that Google's code is an infringing derivative work of Oracle's specifications because the Android source code implements the APIs described by those specifications. But it is again unclear what Oracle means in its references to "interrelationships" and "layout" with respect to the Android source code. And, to the extent that "layout" refers to aspects of the visual layout of the specifications, it is unclear how that applies to the source code, which does not share that visual layout.

B. The parties should brief the legal issues relevant to determining copyrightability.

Once Oracle has stated the basis for its claim, Google proposes that the Court order the parties to brief the following issues that related to copyrightability.

1. Whether the selection, arrangement and structure of the APIs described by the 37 API specifications are ideas, systems or methods of operation.

Ideas, systems and methods of operation are not protected by copyright. 17 U.S.C. § 102(b). Google recognizes that the Court has already concluded that the specifications *themselves* (*i.e.*, the "written documentation") are not methods of operation. Copyright MSJ Order (Dkt. 433) at 11. However, even if the specifications are not themselves methods of operation, if the *material allegedly copied from them* are ideas, systems or methods of operation, Oracle cannot rely on that alleged copying to support its copyright claim. This is because "the party claiming infringement may place '*no* reliance upon any similarity in expression resulting from' unprotectable elements." *Apple VII*, 35 F.3d at 1446 (quoting *Aliotti v. R. Dakin & Co.*, 831 F.2d 898, 901 (9th Cir. 1987), with emphasis added by the Ninth Circuit panel in *Apple VII*).

In its prior order, the Court did not address the copyrightability of the selection, arrangement and structure of the API elements described in the 37 API specifications. *See* Copyright MSJ Order at 10. The Court noted that "Google may be trying to head off a possible argument by Oracle that the APIs described in the specifications are nonliteral elements of the specifications subject to copyright protection," but that it was "unclear whether Oracle is advancing such an argument." *Id.* The Court thus concluded that "the issue is not properly teed up for summary judgment" and declined to "decide whether APIs are methods of operation." *Id.*

It now appears that Oracle *is* taking the position that the selection, arrangement and structure of the APIs described by the 37 API specifications have been copied, and indeed that this is the principal basis for its claim that Google's code implementing those APIs is an infringing derivative work. The issue is thus ripe for the Court's adjudication.

Addressing this issue will require the parties to update the briefing and declarations they submitted in connection with Google's copyright summary judgment motion, taking into account the Court's guidance in the Copyright MSJ Order. The parties' prior briefing on the "method of

operation" issue largely focused on legal issues. The only *factual* issues raised by Oracle in its opposition on this point were as follows:

> The *Lotus* case is also factually distinguishable. It concerned a consumer-friendly set of 50 menus with a simple hierarchy. [citation omitted]. This type of simple user menu cannot compare to the Java APIs, which are "designed for programmers and are designed to provide and describe a very rich development environment." (Mitchell Opp. Report ¶ 18.) They contain thousands of elements, layers of complex interdepencies [sic], data structures, and numerous expressive choices that are not dictated by function.

Oracle's Copyright MSJ Opp. (Dkt. 339) at 11.

Google does not dispute that the APIs contain thousands of elements—the "names" that the Court has held are *not* protected by copyright—that are structured in particular ways. The question the Court would need to resolve is whether this numerosity alters the legal outcome based on creativity in the selection, arrangement or structure of those unprotectable elements. Google also does not dispute that, in at least many instances, Sun may have had other choices it could have made when it designed the APIs. Google contends, however, that those choices were about what *ideas, systems or methods of operation* to include in the API packages. The issue for the Court would, again, be a legal issue, not a factual one.

2. Whether the selection, arrangement and structure of the APIs described by the 37 API specifications are functional requirements for compatibility.

In *Sega Enters. Ltd. v. Accolade, Inc.*, the Ninth Circuit noted that "functional requirements for compatibility . . . are not protected by copyright." 977 F.2d 1510, 1522 (9th Cir. 1992) (citing 17 U.S.C. § 102(b)). Thus, creating a platform that is compatible with software written for another platform does not infringe if the implementing code was not copied. *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000). Notably, nothing in the logic of *Sega* or *Sony* suggests that functional requirements for compatibility are unprotected only when used to achieve *full* compatibility with an *entire* platform. Unprotectable elements are unprotectable, regardless of whether some or all of them are used. Moreover, "[p]urely functional items or an arrangement of them for functional purposes are wholly beyond

the realm of copyright as are other common examples of user interfaces or arrangements of their individual elements—the dials, knobs and remote control devices of a television or VCR, or the buttons and clocks of an oven or stove." *Apple V*, 799 F. Supp. at 1023.

Google contends that any similarities in the selection of the API elements and their hierarchical arrangement and structure (and, in all likelihood, their "interrelationships," whatever Oracle might mean by that) are required to ensure compatibility with the API packages described by the 37 API specifications. Indeed, in the words of Oracle's expert Dr. Mitchell, the selection of the API elements and their hierarchical arrangement (and their "interdependencies") are present to "impose a level of abstraction and structure on top of the underlying software development platform." Mitchell Copyright Opp. Report ¶ 23. The selection, arrangement and structure are there because programmers "require some form of order in order to solve complex programming problems and work efficiently." *Id.* "The expression of the API in a particular hierarchical structure, with rules that are consistent, logical and easy to follow," *id.*, serves the functional purpose of promoting effective programming, just as the arrangement of dials and knobs of a television or VCR serve functional purposes. *See Apple V*, 799 F. Supp. at 1023.

These facts are not reasonably subject to dispute. As explained in the prior paragraph, Oracle's own expert has described the functional purposes of the selection, arrangement and structure of the 37 API specifications. And as Google's expert Dr. Astrachan has explained,

> in order for existing code in a language to be compatible and interoperable with new software written in the same language, the API elements that constitute the language *must also be present, and named and organized identically.*

Astrachan Opening Report (Dkt. 262-1) ¶ 131 (emphasis added). Oracle's expert argues that Android is incompatible with Oracle's Java platform *as a whole*, because Google did not copy *enough.* Mitchell Copyright Opp. Report (Dkt. 341-2) ¶¶ 106-07, 111-13. But nowhere does he dispute (because he cannot) that to ensure compatibility *with the API packages that Google did implement in Android*, Google *had* to use the same selection of API elements, and the same

arrangement and structure for those elements. That is, regardless of whether Android is compatible with the Oracle API packages that Google *did not* implement, the selection, arrangement and structure of the elements of the API packages Google *did* implement are functional requirements for compatibility *with those API packages.*

Not once is this disputed by Oracle's expert. Dr. Mitchell notes that Google implemented only "51 of 130 high-level packages encompassing numerous Java Class Library APIs . . . ." Mitchell Copyright Opp. Report ¶ 106. He explains that this means that Android is not compatible with the Oracle libraries *that Android did not implement. See id.* ¶ 107 ("For example, Java application developers may want and expect to use java.awt.color, java.lang.management, java.rmi, and other examples but Google chose not to support these in Android."). He further points out that Android includes libraries that are not part of Oracle's Java platform, which means that applications written for Android *that use these additional libraries* will not run, unmodified, on Oracle's Java platform. *See id.* ¶¶ 111-13. Yet neither Oracle nor its expert ever takes issue with the fact that in order to ensure compatibility *with the Oracle API packages that Google did implement in Android*, Google *had* to use the same selection of API elements, and the same hierarchical arrangement and structure for those elements, as described in the 37 API specifications. Instead, Oracle only argues that "[w]ith the exception of a very few classes, the Java APIs are not required to use *Java* at all." Oracle's Copyright MSJ Opp. at 20 (emphasis added) (citing Mitchell Copyright Opp. Report ¶¶ 57-60). That misses the point. The question of functional compatibility does not depend on whether the APIs described in the 37 API specifications must be implemented in order to use the *Java language* (a question that hinges largely on what one understands "the Java language" to mean). Rather, the question is whether the APIs must be implemented in order *to ensure compatibility with code that uses those APIs.* And the answer to that question is "yes."

As part of these advance determination proceedings, the key points that the Court would need to resolve are (1) whether functional requirements for compatibility with the APIs described by the 37 API specifications are unprotectable (a pure question of law), and, if so, (2) which of the allegedly copied elements of the selection, arrangement and structure are functionally required for compatibility with the APIs (a question of law with subsidiary fact questions that are not reasonably subject to dispute). If the Court has questions regarding any subsidiary facts after reviewing the parties' briefs, the Court could order supplemental briefing or schedule an evidentiary hearing at which the parties' experts would testify.

3. Whether the selection, arrangement and structure of the 37 API specifications have merged with the underlying ideas expressed.

When an idea "can only be expressed in so many ways," the expression has "merged" with the idea. *Apple VII*, 35 F.3d at 1444. In such a situation, "even verbatim reproduction of a factual work may not constitute infringement." *Allen v. Academic Games League of America, Inc.*, 89 F.3d 614, 617-18 (9th Cir. 1996). The doctrine of merger is "particularly applicable with respect to games since they consist of abstract rules and play ideas." *Id.* (quotation marks and citation omitted). The same principle applies here, where the Court has already noted that an API is "the abstract concept of an interface between programs." Copyright MSJ Order at 10. "In order not to confer a monopoly of the idea upon the copyright owner, such expression should not be protected." *Computer Assocs. Int'l, Inc. v. Altai*, 982 F.2d 693, 708 (2d Cir. 1992) (citing *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971)).

In *Allen*, the court held that the plaintiff had "not shown that it is possible to distinguish the expression of the rules of his game manuals from the idea of the rules themselves." 89 F.3d at 618. Here, Google contends that Oracle cannot show that it is possible to distinguish (1) any expression in the 37 API specifications regarding the selection of API elements, or their arrangement and structure, from (2) the ideas of the APIs themselves. Indeed, the selection of

API elements, and their arrangement and structure, are the "rules" for using the APIs, and thus analogous to the game rules that were at issue in *Allen*. Oracle's expert Dr. Mitchell has explained that "an API consists of *a set of names* that can be used to access features of the library, *together with specified conventions* about their use," and that an "API specification . . . describes *a set of rules* that the code implementing the library *must follow*." Mitchell Copyright Opening Report (Dkt. 341-1) ¶¶ 52, 175 (emphasis added).

In order to resolve this issue, the Court will need to consider any arguments that Oracle makes (and any response by Google in opposition) about how any allegedly copied selection, arrangement and structure are distinguishable from the ideas underlying the APIs that the specifications describe.

4. Whether the selection, arrangement and structure of the 37 API specifications are *scenes a faire.*

"Under the scenes a faire doctrine, when certain commonplace expressions are indispensable and naturally associated with the treatment of a given idea, those expressions are treated like ideas and therefore not protected by copyright." *Swirsky v. Carey*, 376 F.3d 841, 850 (9th Cir. 2004). In the computer context, this includes situations in which

> a programmer's freedom of design choice is . . . circumscribed by extrinsic considerations such as . . . (2) compatibility requirements of other programs with which a program is designed to operate in conjunction; . . . (4) demands of the industry being serviced; and (5) widely accepted programming practices within the computer industry.

Computer Assocs., 982 F.2d at 709-10.

Here, Google contends that implementing the APIs described by the 37 API specifications was necessary to ensure compatibility with Java programs that use those APIs ("compatibility requirements of other programs with which a program is designed to operate in conjunction"), demands of the Java programming community ("demands of the industry being serviced"), and accepted Java programming practices ("widely accepted programming practices within the computer industry").

Oracle's expert essentially concedes these points. As discussed above, implementation of the APIs as described by the specifications was necessary to ensure compatibility. With respect to the demands of the Java community and accepted Java practices, although Dr. Mitchell claims that Google did not *need* to implement the APIs, he nowhere disputes that Java programmers *expect* to be able to use the APIs, and that it is *accepted practice* to use the APIs when programming in Java. Indeed, Dr. Mitchell claims that if Google had not adopted the Java APIs, it would have "went to market with an unfamiliar application development environment . . . ." Mitchell Copyright Opp. Report ¶ 60. Far from claiming that the APIs are not demanded by the industry, Dr. Mitchell suggests that Java programmers "may want and expect" Java APIs *in addition* to the ones that Google implemented. *Id.* ¶ 107. Industry demand and accepted programming practices are also evidenced by the fact that the two other major independent Java library implementations (Apache Harmony and GNU Classpath) also implement all of the 37 API specifications, and have done so openly for years. To the extent that Oracle believes these facts to be truly disputed, it can offer declarations in support of its position. After reviewing the parties' submissions, the Court can decide the issue, or, if necessary, order supplemental briefs or schedule an evidentiary hearing.

Regardless, the key issue is a pure issue of law. The Court must decide whether the constraints on Google's choices at the time the alleged infringement began are relevant, or whether the relevant constraints are those that Sun faced at the time the APIs were designed. According to Oracle, the Court must look only at the constraints Sun faced. Copyright MSJ Opp. at 16 (citing *Control Data Sys., Inc. v. Infoware, Inc.*, 903 F. Supp. 1316, 1323 (D. Minn. 1995)). This, however, is not the law in the Ninth Circuit. *See* PAUL GOLDSTEIN, GOLDSTEIN ON COPYRIGHT § 2.3.2.1 (3d ed. 2011) (observing that courts in the Ninth Circuit have held that "it is the range of expressive choice that existed at the time the competing product was created—not the range of expression that existed at the time the copyrighted work was created—that

controls"); *see also Sega*, 977 F.2d at 1524 & n.7 (copying "unlock" code did not infringe, because the *defendant* had no alternatives to using that code).

III. Examples using APIs.

The Court requested that Google "use at least two APIs as examples" to ensure that its proposal is concrete. 10/31/11 Order at 2. Below, Google discusses examples involving the selection of APIs for three classes, one method, and one field that are part of the java.net package. Google also discusses examples involving the hierarchy of the APIs for four classes that are part of the java.net package, and that are subclasses (or in one instance a subclass of a subclass) of the Object class in the java.lang package.

Oracle claims that its selection of elements in the APIs described by the 37 API specifications is protected by copyright. It will presumably identify that selection as part of its infringement case. For example, one of the accused API packages is java.net. Google contends that its implementation of java.net had to include the same selection of elements described in Oracle's specification for the java.net package in order to be compatible with code written in the free and open Java language that relies on the elements that are part of the java.net APIs.

The java.net package includes classes called "Inet4Address," "Proxy," and "Sockets" (and many other classes; these are merely three examples). Google declarants can testify that unless Google implemented those classes, Android would not be compatible with code written in the Java language that uses java.net and those classes.

Similarly, Google's declarants can testify that unless Google implemented the methods that are part of those classes (such as the "getAddress" method that is part of the Inet4Address class), and their fields (such as the "NO_PROXY" field that is part of the Proxy class), Android would not be compatible with code written in the Java language that uses those methods and fields. The Google declarants can also testify that implementing the "interfaces," "enums," and "exceptions" that are part of the java.net package was necessary to ensure compatibility with

code written in the Java language that relies on those elements.

Oracle also claims that the hierarchy of the elements in its APIs has been copied. For example, Oracle will presumably identify the hierarchy of classes and subclasses as part of its infringement case. The beginning of the hierarchy of the classes in the java.net package, as illustrated in Oracle's expert's opposition report, is as follows:

- java.lang.**Object**
  - java.net.**Authenticator**
  - java.net.**CacheRequest**
  - java.net.**CacheResponse**
    - java.net.**SecureCacheResponse**

See Mitchell Copyright Opp. Report ¶ 30. This means that the Authenticator, CacheRequest and CacheResponse classes that are part of the java.net package are all "subclasses" of the Object class that is part of the java.lang package, and that the SecureCacheResponse class that is part of the java.net package is a subclass of the CacheResponse subclass.

Google declarants can testify that if Google did not implement the same class hierarchy when it implemented the java.net package, code written in the Java language that relies on the "inheritance" characteristics of these classes would not be compatible with Android. Google declarants can offer similar testimony about the other hierarchical features of the APIs described by the 37 API specifications.

Google does not believe that Oracle can reasonably dispute any of these facts. Oracle, however, would be entitled to attempt to rebut Google's evidence, or to argue that these facts do not render the selection, arrangement and structure of its API specifications uncopyrightable.

Google can offer similar testimony about the selection, arrangement, and structure of the other accused API specifications, and the classes, subclasses, interfaces, fields, methods and exceptions that are part of the packages described in those specifications. Because there are thousands of elements described in the 37 API specifications, and because the arrangement and structure of those elements is similarly voluminous, in all likelihood the Google declarants would

begin with specific examples, such as the ones noted above, and then testify more generally that the same is true of the other accused API specifications, and the classes, subclasses, interfaces, fields, methods and exceptions that are part of the packages described in those specifications.

Moreover, although Google is not clear what precisely Oracle means when it refers to allegedly copied "interrelationships," Google suspects that these "interrelationships" are also functionally required for compatibility. If so, Google declarants can offer similar testimony about the "interrelationships." And, depending on what Oracle means when it refers to allegedly copied "layouts," Google declarants can also offer similar testimony about the "layouts."

Google recognizes that the evidence it proposes to offer may sound tautological. That is, the declarants would testify that various selections, structures and arrangements had to be implemented in the same way to ensure compatibility with code that relies on those selections, structures and arrangements. But this is precisely the *point* of the specifications. The specifications state what behaviors and characteristics programmers can assume that the packages described in those specifications have. In order to ensure compatibility with the APIs described in those specifications, Google *had to* implement the behaviors and characteristics described in those specifications. Google declarants can also testify to that effect. Indeed, as Google's expert has already testified,

> If these elements are not present and identical in different implementations of the same API, programmers will not be able to use the same names and structures when using the API, since it is these elements that allow each piece of the software to speak to each other. If these names or structures are changed, software that references these names will fail to function, because the software will not be able to find and access the functionality it needs.

Astrachan Opening Report ¶ 53. Nowhere has Oracle or its expert disputed any of this.

Indeed, perhaps the simplest example showing what results if APIs are not implemented, or are implemented differently, comes from Oracle's expert Dr. Mitchell's explanation of Java user interface APIs that Google did *not* implement. As Dr. Mitchell explains, in Oracle's Java platform, "this functionality is provided by the APIs for AWT . . . and Swing . . . ." Mitchell

Copyright Opp. Report ¶ 63. Google did not implement these APIs, instead designing new user interface APIs. *Id.* ¶¶ 63-64. Dr. Mitchell wrote code using the Oracle and Google APIs that performed the same functions. *See id.* ¶ 69. This is the code using the Oracle Java APIs:

```
JOption.showMessageDialog(frame, "Do you like green eggs and
ham?", "Message");
```

Id. ¶ 66. The code he wrote using the different Android APIs is completely different:

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Do you like green eggs and ham?")
       .setNegativeButton("OK", new
DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
           }
       });
AlertDialog alert = builder.create();
. . .
showDialog(ID);
```

Id. ¶¶ 67-68. In short, implementing different APIs, or implementing them differently, requires the programmers using the APIs to write different code. When the same APIs are not implemented in the same way, the results are not compatible.

If the Court has questions about any of these technical concepts, it could schedule a tutorial session. *See Apple V,* 799 F. Supp. at 1017 n.3 (discussing the tutorial held with Judge Walker and how it assisted the court). Alternatively, in light of the Court's limited time, the Court might wait until after briefing, and then, if necessary, order the parties' technical experts to answer any questions the Court might have, either by written response or live testimony.

IV. Determining copyrightability now could dramatically simplify trial.

If the Court were to conclude that some or all of the selection, arrangement and structure of the 37 API specifications are not copyrightable, the copyright claim would be substantially narrowed. This would simplify and shorten trial of the remaining copyright issues.

Sincerely,
/s/
Robert A. Van Nest